

Responzivni web dizajn

Hečimović, Helena

Undergraduate thesis / Završni rad

2018

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Organization and Informatics / Sveučilište u Zagrebu, Fakultet organizacije i informatike**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:211:003474>

Rights / Prava: [Attribution-NonCommercial-NoDerivs 3.0 Unported](#) / [Imenovanje-Nekomercijalno-Bez prerada 3.0](#)

Download date / Datum preuzimanja: **2024-07-23**



Repository / Repozitorij:

[Faculty of Organization and Informatics - Digital Repository](#)



**SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
V A R A Ź D I N**

Helena Hećimović

RESPONZIVNI WEB DIZAJN

ZAVRŠNI RAD

Varaždin, 2018.

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
V A R A Ź D I N

Helena Hećimović

Matični broj: z-41528/12-izv

Studij: Primjena informacijske tehnologije u poslovanju

RESPONZIVNI WEB DIZAJN
ZAVRŠNI RAD

Mentor :

Prof.dr.sc Dragutin Kermek

Varaždin, rujan 2018.

Helena Hečimović

Izjava o izvornosti

Izjavljujem da je moj završni/diplomski rad izvorni rezultat mojeg rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

Autor/Autorica potvrdio/potvrdila prihvaćanjem odredbi u sustavu FOI-radovi

Sažetak

Ovaj završni rad obrađuje temu respozivnog web dizajna, njegovih prednosti i nedostataka. Osim respozivnog web dizajna, pozornost je usmjerena na suvremene i temeljne tehnike i tehnologije poput HTML5, CSS3, JavaScript, PHP/SQL, Bootstrap-a te povijest weba i CMS (Content Management System).

Praktični dio bazira se na izradu web aplikacije „Apartmani Zaton“. Posebna pozornost na izradu prikaza fotogalerije i njezinu prilagodbu različitim veličinama ekrana. Izvršena je analiza već postojećih web lokacija o apartmanima kako bi se postavili temelji i usporedbe za praktični dio rada. Sveukupan izgled web aplikacije odabrati će se analizom suvremenih vizualnih trendova. Nakon svake izvršene faze, autor analizira stanje te mijenja dizajn po potrebi.

Sadržaj

| | |
|----------------------------------------------|----|
| 1.Uvod..... | 1 |
| 2.Web dizajn..... | 2 |
| 2.1.Općenito..... | 2 |
| 2.1.1. Jezici za izradu web stranica..... | 3 |
| 2.1.1.1. HTML..... | 3 |
| 2.1.1.2. CSS..... | 6 |
| 2.1.1.3. JavaScript..... | 7 |
| 2.1.1.4. PHP/SQL..... | 8 |
| 2.2.Responzivni web dizajn..... | 10 |
| 2.2.1. Karakteristike..... | 10 |
| 2.2.2. Prednosti/Nedostaci..... | 12 |
| 2.2.3. Bootstrap..... | 13 |
| 2.3.Povijest web dizajna..... | 14 |
| 2.4.CMS..... | 15 |
| 3.Praktični dio..... | 16 |
| 3.1.Opći prikaz web aplikacije..... | 16 |
| 3.1.1.Dizajn zaglavlja..... | 17 |
| 3.1.2.Bootstrap..... | 18 |
| 3.1.3.Dizajn navigacije..... | 20 |
| 3.1.4. Parallax..... | 23 |
| 3.1.5.Sadržaj stranice..... | 24 |
| 3.1.6.Slideshow..... | 27 |
| 3.1.6.1.HTML..... | 27 |
| 3.1.6.2.JavaScript..... | 28 |
| 3.1.6.3.Konačan izgled slideshow-a..... | 29 |
| 3.1.7.Google Map..... | 29 |
| 3.1.8.Podnožje..... | 33 |
| 3.1.9.Podstranice..... | 34 |
| 3.2.Postavljanje na server i testiranje..... | 37 |
| 4.Zaključak..... | 39 |
| Literatura..... | 40 |
| Popis slika..... | 41 |

1. Uvod

„Apartmani Zaton“, kao turistički smještaj, su u funkciji već 13 godina. Cilj iznajmljivanja smještaja u Zatonu je imati što više rezervacija i zadovoljnih turista. Uz lijepo i moderno uređene objekte, tu su još i brojne pogodnosti, npr. Velik izbor plaža, pristup biciklima, mala udaljenost trgovina i pekara, mirno susjedstvo itd. Kako apartmani nemaju vlastitu web stranicu, potrebno je izraditi istu i omogućiti vlasnicima još bolje oglašavanje koje će biti u skladu s trendovima i konkurencijom i zadovoljavati web standarde.

Teorijski dio obrađuje temu web dizajna općenito, njegovo razvijanje kroz povijest te jezike koji su potrebni da bi se uopće realizirala web aplikacija. Obrađena je tema samog responzivnog web dizajna, njegove karakteristike te prednosti i nedostaci korištenja istog. Opisan je sustav koji omogućuje upravljanje sadržajem, odnosno CMS (Content Management System). Spomenut je jedan web programski okvir otvorenog koda, Bootstrap, koji je trenutno i jedan od najpopularnijih web programskih okvira.

Praktični dio bavi se razvojem i oblikovanjem web aplikacije za već spomenute „Apartmane Zaton“, čiji je glavni cilj postizanje veće informiranosti turista o njihovoj ponudi i sadržaju. Objašnjene su sve faze izrade web aplikacije, od izrade vizualnog prototipa do razvoja pomoću različitih dostupnih tehnologija.

2. Web dizajn

Web dizajn je pojam koji obuhvaća različite standarde, vještine i discipline koje se koriste u izradi web stranica. Web dizajn uključuje procese istraživanja, definiranja i strukturiranja sadržaja, izradu dizajna, programiranje, unos sadržaja te testiranje i objavu na internetu. Pomoću web dizajna se planiraju, izrađuju i razvijaju web stranice. Taj proces izrade uključuje dizajniranje izbornika za navigaciju, dizajn teksta, prilagodba naslova, boje, fontove i slike.

2.1. Općenito

Web dizajn uključuje različite vještine i discipline u izgradnji i održavanju web stranica. Pojam web dizajn se obično koristi kako bi se opisao postupak dizajniranja, koji se odnosi na korisničku stranu (eng. front-end) konstruiranja web stranice uključujući pisanje kôda. Web dizajner je osoba koja se bavi izradom web stranica primjenom najnovijih tehnika i tehnologija. Za izradu web dizajner koristi standardne programske alate koji se koriste za kreiranje i formatiranje web dokumenata kao što su HTML, CSS, JavaScript. Specifične programske alate za programiranje na korisničkoj i poslužiteljskoj strani, administriranje baze podataka i slično.

Web se neprekidno mijenja stoga se od web dizajnera očekuje da bude uvijek u tijeku sa trenutnim događanjima, odnosno trendovima koji su prisutni. Dobar web dizajner i web programer mora biti uvijek dobro informiran, pratiti kako se stvari odvijaju u sadašnjici u području programiranja za web. Dizajner korisničkih iskustava (eng. User experience designer), dizajnira proizvode koji su korisni i ugodni za interakciju. Radi se o povećanju iskustva između korisnika i kreiranog proizvoda. Jedan od glavnih zadataka dizajnera korisničkih iskustava je kakav ugođaj korisnik ima kada je u doticaju sa proizvodom. Dizajner korisničkog sučelja, spaja interakciju, vizualni dizajn i informacijsku arhitekturu, također može izgraditi sučelje koje funkcionira na različitim uređajima.

Prije samog početka dizajniranja, potrebno je definirati i deklarirati grid, odnosno mrežni sustav (grid system). Grid (rešetka) je korisna pri strukturiranju različitih sekcija web stranice i one su osnovni element dizajna, te vodi dizajnera kroz specifične veličine ekrana. Mreža web dizajnerima olakšava postavljanje pojedinih elemenata kroz pisanje kôda. Nakon što je deklarirana mreža s elementima, prelazi se na određivanje boja. Pri odabiru boja, koji je jedan od veoma bitnih elemenata dizajna, važno je da budu zadovoljene teorijske osnove web dizajna. Također, jednako je važno uzeti u obzir kakav će sadržaj biti na web stranici odnosno o čemu je na web stranici riječ. Odabirom boje dizajner određuje koje će emocije prevladavati na web

stranici. Tekst na web stranici trebao bi biti čitljiv i nenametljiv. Također vodi se računa i o naslovima. Poštujući standarde i trendove osigurava se da su stranice dostupne širokom spektru korisnika.

2.1.1. Jezici za izradu web stranica

Ovisno o njihovim funkcijama, razlikujemo nekoliko podjela programskih jezika. Osnovna podjela je na korisničku stranu i pozadinski sustav (engl . Backend). Jezici koji se koriste za programiranje na korisničkoj strani interpretiraju se unutar web preglednika i za prikaz im nije potreban poslužitelj. U toj kategoriji nalaze se HTML, CSS i Javascript. Pozadinski jezici izvršavaju se na poslužitelju te rezultat šalju web pregledniku, odnosno krajnjem korisniku. Podrazumijevaju rad na arhitekturi i logici sustava. Najpopularniji su PHP, SQL, ASP.

2.1.1.1. HTML

Hypertext Markup Language (HTML) je standardni opisni jezik oznaka za izradu web stranica i web aplikacija. 1989. godine razvio ga je Tim Berners-Lee. Trenutno aktualna verzija je HTML5. HTML je jezik za označavanje podataka koji se rabi za opis web dokumenta. Takve jezike obilježava korištenje oznaka (engl. Tag), za opisivanje sadržaja dokumenta. HTML dokumenti šalju se s poslužitelja prema klijentima. Klijent preuzima HTML dokument te njegovu interpretaciju prikazuje korisniku na ekran. Svaki HTML dokument treba posjedovati minimalnu strukturu koja je potrebna za pravilno interpretiranje sadržaja. Pod minimalnim dokumentom se podrazumjeva onaj dokument koji prolazi metode validacije HTML koda te se može prikazati na ekranu nekog web preglednika. Minimalna HTML5 stranica bi trebala izgledati ovako:

```
1.<!DOCTYPE html>
2.<html>
3.<head>
4.    <meta charset="UTF-8">
5.    <title> Naslov </title>
6.</head>
7.<body>
8.    <p> Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam
cursus. Morbi ut mi.
9.        Nullam enim leo, egestas id, condimentum at, laoreet mattis,
massa. Sed eleifend
10.        nonummy diam. </p>
11. </body>
12. </html>
```

<!DOCTYPE html> Predstavlja deklaraciju dokumenta. Funkcija je da web pregledniku kaže kakav dokument prima na čitanje. Ova deklaracija govori da je riječ o HTML5 dokumentu.

<html> ... </html> Oznaka *html* rabi se za označavanje početka i kraja HTML sadržaja. Unutar njega se nalaze deklaracije svih skripti, vanjskih resursa i sadržaja kojima se dokument koristi da bi prikazao tražene informacije.

<head> ... </head> Oznaka *head* sadržava metainformacije o dokumentu.

<body> ... </body> Oznaka *body* unutar sebe sadrži sadržaj koji se korisniku prikazuje unutar preglednika.

<title> Naslov </title> Sadržava naslov dokumenta.

Za validni su dokument nužni najmanje deklaracija tipa dokumenta i oznaka *title*.

HTML posjeduje svoje oznake koje se nazivaju HTML oznake. Svaka oznaka opisuje svoj dio dokumenta. Osnovni HTML oznake tvore se na sljedeći način:

<! – osnovna oznaka -->

Tablice služe za prikaz podataka na jasan način, u nastavku slijedi primjer HTML tablice:

```
<!DOCTYPE html>
  <html>
    <head>
      <title> Tablica </title>
    </head>
    <body>
      <table border="1">
        <caption> Popis osoba </caption>
        <tr>
          <th>Ime</th>
          <th>Prezime</th>
          <th>Dob</th>
        </tr>
        <tr>
          <td>Helena</td>
          <td>Hečimović</td>
          <td>24</td>
        </tr>
        <tr>
          <td>Iva</td>
          <td>Ivić</td>
```

```
<td>45</td>
</tr>
</table>
</body>
</html>
```

Na slici je prikazan rezultat prethodnog koda, odnosno prikaz dobivene tablice.



| Ime | Prezime | Dob |
|--------|-----------|-----|
| Helena | Hečimović | 24 |
| Iva | Ivić | 45 |

Slika 1. Rezultat primjene HTML koda za tablicu

HTML poveznice (engl. Anchor) rabe se za stvaranje elemenata na koje se može kliknuti unutar web stranice te koji korisnika vode na novu lokaciju. Poveznice rade tako da se unutar atributa *href* smjesti adresa odnosno lokacija do koje se treba doći. Poveznica može biti definirana kao tekstualna poveznica ili kao slika poveznica.

Poveznice mogu voditi na druge lokacije na tri načina :

- Apsolutna putanja
- Relativna (lokalna) putanja
- Putanja unutar stranice

2.1.1.2.CSS

CSS (Cascading Style Sheets) je stilski jezik kojim se opisuje izgled dokumenta napisanog u HTML-u. CSS opisuje kako bi HTML elementi trebali biti prikazani u web pregledniku. CSS se koristi za definiranje stilova sadržaja web stranice, uključujući dizajn, izgled, fontove, za različite uređaje i veličine zaslona. CSS ima mnogo veće mogućnosti za razliku od oblikovanja stranica samo HTML elementima. Definiranje stila obično se izvršava u posebnoj, vanjskoj .css datoteci. S vanjskom datotekom u kojoj se definira stil HTML dokumenta, može se promijeniti izgled cjelokupne web stranice. CSS je moguće pisati na tri različita načina:

- Unutar HTML dokumenta.
- Inline CSS : piše se unutar HTML oznaka.
- Vanjski CSS : zasebni dokument s ekstenzijom *.css koji se poziva u oznaci head.

CSS3 je najnovija specifikacija jezika *Cascading Style Sheets* i cilj joj je proširiti CSS2 verziju. CSS3 donosi mnogo, dugo očekivanih noviteta, kao što su zaobljeni uglovi, sjene, fleksibilni izgledi okvira i rešetke i mnoge druge.

CSS3 je specifikacija koja se razvija. Razvojni timovi koji rade na preglednicima je implementiraju, a zajednica programera nadograđuje, proučava i piše o njoj. Neki noviteti koje je donio CSS3:

- Animacije
- Bolja kontrola tipografije
- Obli rubovi
- Sjene
- Višestruki stupci

Liste stilova čine selektori koji sadrže listu svojstava i vrijednosti koje se pridružuju svojstvima. Sva svojstva sadržana u jednom selektoru definiraju se između vitičastih zagrada ({ , }). Unutar jednog selektora može se definirati proizvoljan broj svojstava, a sva se svojstva međusobno razdvajaju točkom sa zarezom (;). Bilo koji HTML element se može uporabiti kao selektor, a

svojstva oblikovanja definirana unutar selektora primjenjuju se na sadržaj koji zaokružuje taj element. Sintaksa selektora definirana je na sljedeći način:

```
selektor {  
  
    svojstvo_1: vrijednost_1;  
    svojstvo_2: vrijednost_2;  
    ...  
  
}
```

Ukoliko se žele primjeniti ista svojstva na više različitih HTML elemenata, elementi se odvajaju zarezom. Na primjer, želi li se primjeniti crvena boja teksta za numerirane, nenumerirane liste i odlomke, selektor će se definirati kao na sljedećem primjeru:

```
p, ul, ol { color: red; }
```

Ako se želi primjeniti svojstvo oblikovanja samo na ugniježdene elemente, selektor se treba definirati na sljedeći način :

```
p ul { color: red; }
```

2.1.1.3.JavaScript

Treći stup razvoja Web lokacija, pored strukture i oblikovanja, jest funkcionalnost za koju je na Webu zadužen JavaScript. Javascript je jedan od najbrže rastućih programskih jezika. JavaScript je objektno orijentirani programski jezik skriptiranja. Predstavljen je 1995. godine kao LiveScript. Ubrzo nakon toga je promijenjen naziv u JavaScript. Postoje tri razine JavaScript jezika, a to su:

- Središnji dio
- Programiranje na strani klijenta
- Programiranje na strani poslužitelja

Za Javascript je stvoren niz programskih okvira (eng. Framework), kako bi se programerima olakšao posao. Jedan od najpoznatijih je jQuery. JavaScript programski okviri su kolekcije već

napisanih skripti čiji je cilj smanjiti troškove razvoja aplikacija. Pisanje vlastitih skripti ponekad je najbolji pristup, posebno ako se želi izbjeći velike datoteke pojedinih programskih okvira. Ali, ako se radi u velikim timovima, gdje se traži da svi rade prema istim standardima ili kada se treba brzo razviti rješenje, preporuka je koristiti programske okvire. Da bi se koristio programski okvir, mora se samo preuzeti kopija programskog okvira ili zadati vezu do nje, a sav kod koji se oslanja na programske okvire smjestiti u datoteke koje slijede. JQuery pojednostavljuje način na koji se pišu skripte. Funkcionira tako da raspolaže skupom metoda koje od autora traže jednostavne argumente, a obavlja prilično komplicirane poslove. Jedna odlična značajka programskog okvira jQuery jest mogućnost nizanja metoda uz miješanje selektora i akcija kako bi se izradili dugački iskazi.

2.1.1.4.PHP/SQL

PHP je nasljednik starijeg projekta koji je razvio Rasmus Lerdorf 1995. godine, a zvao se PHP/FI. U početku je jezik PHP bio zamišljen kao skup skripti Perl koju je Rasmus napravio kako bi mogao pratiti posjete korisnika njegovu *online* životopisu. Taj je skup skripti nazvao „Personal Home Page Tool“. No, kako su rasli zahtjevi za funkcionalnošću, napisao je mnogo veću C implementaciju koja je imala mogućnosti komuniciranja s bazom podataka i omogućavala je korisnicima razvijanje jednostavne dinamičke web stranice ili aplikacije. Odlučio je objaviti svoj kod kako bi se njime svi mogli koristiti, ali i kako bi mu pomogli pri ispravljanju grešaka. PHP/FI jest akronim kojni označava Personal Home Page/Forms Interpreter, a uključuje neke osnovne funkcionalnosti koje su i danas sastavni dio PHP-a. Druga se inačica PHP/FI-ja pojavila 1997. I već je tada imala više od nekoliko tisuća sljedbenika u cijelom svijetu.

PHP 3 jest prva inačica PHP-a kakvog danas poznajemo. Stvorila su ga dvojica programera, Andi Gutmans i Zeev Suraski, nakon što su shvatili da PHP/FI 2.0 nije dovoljno moćan za njihov projekt na sveučilištu. Napisali su ga potpuno iz početka i objavili kao nasljednika PHP/FI-ja 2.0. Njegova je najjača osobina mogućnost upotrebe ekstenzija za proširenje osnovne funkcionalnosti. Ovo je omogućilo priključenje novih programera na razvoju ekstenzija. Tako programeri nisu morali pisati cijeli kod ispočetka, već samo ekstenziju za ono što im treba, kao što je recimo spajanje na bazu ili nekakav API za komunikaciju. Jednako tako, velika novost u PHP-u 3.0 bilo je uvođenje objektno orijentirane sintakse. Odmah je postao popularan te je zabilježeno da je do kraja 1998. Bio instaliran na 10% svih svjetskih web poslužitelja. Nedugo nakon izlaska PHP-a 3.0., Andi Gutmans i Zeev Suraski počeli su raditi na inačici 4.0. Zapravo su ponovno prepisali jezgru sustava kako bi omogućili još veću povezanost s novim ekstenzijama te kako bi dobili još bolje performanse. Tu su novu jezgru nazvali „Zend Engine“

kao izvedenicu njihovih imena i predstavili je sredinom 1999. Zadovoljili su sve svoje ciljeve proširivosti i performansa te učinili ovu platformu još popularnijom i privukli još veći broj programera koji rade na unaprjeđenju koda. PHP 5 predstavljen je sredinom 2004. Njegova je glavna novost nova jezgra „Zend Engine 2.0“ koja objedinjuje novi modul za objektno orijentirano programiranje te još nekoliko novih funkcionalnosti.

Za razvoj PHP aplikacije potrebno je upogoniti poslužitelj Apache. Apache se bavi interpretiranjem dokumenata prije nego što dođu do korisnika te izvršava sav PHP kod na koji naiđe pri čitanju. Kod izrade web aplikacija i stranica u PHP-u, osim aplikacije za pisanje koda i preglednika, potreban je i web poslužitelj. Web poslužitelj je aplikacija koja je instalirana na poslužitelju na kojem je smještena naša stranica. Služi za izvođenje programskog koda i prikazivanja rezultata korisniku u web pregledniku. Najpopularniji web poslužitelj danas je Apache. PHP kod pohranjujemo u datoteke s ekstenzijom .php i s pomoću FTP programa smještamo ih na naš poslužitelj. Kao što HTML dokument korisniku može pokazati da je u pitanju HTML, tako i PHP posjeduje svoje oznake koji pokazuju početak odnosno kraj PHP dokumenta. PHP oznake izgledaju ovako:

```
<?php
```

```
//kod za izvođenje
```

```
?>
```

Oznakom <?php označava se početak PHP skripte. Sve što slijedi nakon toga smatra se PHP kodom.

SQL (Structured Query Language) omogućuje pristup i manipulaciju bazom podataka. SQL izrazi se koriste za obavljanje zadataka poput ažuriranja, brisanja i preuzimanja iz baze podataka. Neki od sustava za upravljanje bazama podataka koji koriste SQL su: Oracle, Microsoft SQL, MySQL, MS Access... MySQL je sustav baze podataka koji se koristi na webu, MySQL radi na poslužitelju i idealan je za male i velike aplikacije. Brz je pouzdan i jednostavan za korištenje, koristi standardni SQL. Baze podataka korisne su za pohranu podataka kategorizirano. Podaci su pohranjeni u tablicama. Tablica je zbirka povezanih podataka, a sastoji se od stupaca i redaka.

2.2.Responzivni web dizajn

Tokom vremena, promjenili su se načini na koje pristupamo webu, prvenstveno uređaji s kojih pristupamo webu. Više nema ograničenja korištenjem samo stolnog računala ili koristeći prijenosnik. Danas su tu i mnoge druge kategorije uređaja, primjerice pametni mobiteli, tableti, igrače konzole, čitač e-knjiga... Ponuda uređaja koji mogu pristupati webu je ogromna i svakim danom sve veća. Uz ovaj raspon web uređaja dolazi i nevjerovatna prilika za web dizajnere. U početku je najveći problem responzivnog dizajna bio Internet Explorer i ostali zastarjeli preglednici čije verzije ne podržavaju HTML5 i CSS3. Stoga se često izrađivala zasebna stranica za mobilne preglednike, a zasebna za preglednike na računalima ili se pomoću JavaScripta pokušavalo riješiti problem prilagodbe sadržaja. Srećom, danas se broj korisnika IE preglednika smanjuje jednakom brzinom kojom raste broj korisnika koji stranice posjećuju preko mobilnih uređaja pa i taj problem postaje dijelom prošlosti.

2.2.1.Karakteristike

Tehnologija donosi sve više uređaja koji mogu pristupati webu, kao što su monitori s velikim zaslonima, tableti, pametni telefoni, pametni televizori. Ovaj napredak u mobilnim tehnologijama i brz razvoj sustava za navigaciju kroz web stranice i tehnika prikaza primorao je web dizajnere i web programere da polako zaborave koncept fiksnih dimenzija web stranica i počnu razmišljati o strukturi koja se može adaptirati i koja će ponuditi odgovarajući sadržaj za svaku situaciju. Responzivni web dizajn ili RWD je tehnika koja olakšava prilagodbu web stranica različitim veličinama ekrana, odnosno rezolucijama prikaza. Smisao responzivnog web dizajna je korisniku pružiti jednako kvalitetno iskustvo pregleda web stranice, neovisno o tome kakav uređaj koristi. Tehnički govoreći, to podrazumijeva korištenje ove tri ključne tehnologije:

- Fleksibilan raspored temeljen na mreži
- Fleksibilne slike i videozapisi
- Korištenje CSS-a na takav način da može kanalizirati ponašanje web stranice sukladno uređaju

Kad se radi s responzivnim dizajnom, mora se razmišljati o fluidnosti, adaptaciji, a ne o tome da svaki piksel bude na svom mjestu. Zbog toga se praksa provjeravanja svakog piksela sve rjeđe koristi. Postoje dva načina da se ovaj problem riješi i da web lokacija ostane responzivna:

- Pretvaranje web lokacije korištenjem nešto malo matematike kako bi se osigurao dobar rezultat

- Korištenje sustava responzivne mreže kada se odabere skupina stupaca i kod se zadržava unutar njih korištenjem relativnih dimenzija

Za dizajn suvremenih web lokacija koriste se dvije metodologije: adaptivni web dizajn i responzivni web dizajn. I jedan i drugi dijele kocepte korištenja točaka prekida, što su granice definirane korištenjem upita o mediju na kojima se mijenja raspored stranice. Razlika između adaptivnog i responzivnog pristupa je u ponašanju lokacije između točaka prekida. Adaptivni pristup zapravo je niz rasporeda fiksne širine, dok responzivni pristup koristi fleksibilne dimenzije pa lokacija ostaje fluidna čak i između točaka prekida. Korištenje postotaka umjesto fiksnih vrijednosti znači da se elementi stranice povećavaju ili smanjuju zajedno i sukladno s promjenama prostora za prikaz, pri čemu sadržaj teče između granica zaslona, otuda i naziv fluidan. Ali isto tako, rad s postocima ne teče potpuno glatko, jedan od problema su poteškoće s miješanjem jedinica dužine.

Responzivni web dizajn bazira se na upotrebi CSS-a, odnosno Media Query funkcionalnosti. CSS već neko vrijeme raspolaže načinom da različitim tipovima medija isporuči različite stilove oblikovanja korištenjem atributa *media* elementa. Media Query je modul koji proširuje tipove medija tako što pruža sintaksu upita, koja omogućava isporuku vrlo specifičnih stilova korisnikovom uređaju, odnosno precizno napravljenu prezentaciju sadržaja. Upiti o mediju pružaju slobodu da se konstruiraju web lokacije koje su doista neovisne o uređaju i pružaju korisnicima najbolje moguće iskustvo u korištenju, bez obzira na koji način im pristupaju. Također medijski upiti se mogu koristiti za promjenu veličine fonta elementa na različitim veličinama zaslona.

Jedan od bitnih faktora u responzivnom dizajnu je fluidno skaliranje slika i videa tako da odgovaraju širini ekrana. Jedan od načina na koji se može postići da se slika skalira ovisno o širini ekrana je sljedeći: potrebno je kreirati `<div>` element unutar kojeg se nalazi `` element. U CSS se tada uvede `div` selektor kojemu se zadaje željena širinu i `img` selektor kojemu se definira maksimalnu širinu pomoću `max-width` svojstva čiju vrijednost treba postaviti na 100% te definirati i visinu pomoću `height` svojstva čiju vrijednost treba postaviti na `auto`. Tada slika

zauzima 100% svog roditelja, a kada se njegova širina promjeni slika se skalira ovisno o njegovoj širini. Visinu se postavlja na auto zbog očuvanja omjera visine i širine slike.

Drugi jako bitan faktor je tipografija jer tekst definiran u pikselima na različitim uređajima izgleda različito, a često postaje i nečitljiv. Recept za rješavanje ovog problema mnogo dizajnera pronalazi u upotrebi rem mjernih jedinica. Pomoću njih veličina fonta je relativna u odnosu na glavnu definiranu veličinu fonta. EM mjerne jedinice čine sličnu stvar, ali ovise o roditelju. Unutar html selektora definira se font-size svojstvo na željenu vrijednost (npr. 12px). Ostatak teksta na stranici može se mijenjati pomoću raznih selektora tako da im se font-size svojstvo definira pomoću rem mjernih jedinica (npr. `p {font-size: 1rem;}` bi značilo 28 da su svi paragrafi na stranici veliki 12 piksela). Ukoliko se u media queries doda drugačiju vrijednost tada će jednak tekst na mobilnom uređaju biti npr. velik 2rem čime povećavamo čitljivost na malim ekranima.

2.2.2. Prednosti/Nedostatci

Svrha responzivnog dizajna je imati jedno mjesto, ali s različitim elementima koji različito reaguju kada se gledaju na uređaje različitih veličina. S porastom broja pametnih telefona, tableta i drugih mobilnih računalnih uređaja, više ljudi koristi manje zaslone za pregled web stranica. Korištenje responzivnog web dizajna ima svoje prednosti, neke od njih su više mobilnog prometa, brži mobilni razvoj po nižim cijenama, potreba za održavanjem, brže web stranice, poboljšano iskustvo pregleda na mreži i bez mreže. Sve je važnije da tvrtke imaju web stranice koje se ispravno prikazuju na manjim zaslonima kako korisnici ne bi doživjeli iskrivljene slike ili doživjeli nepregledan izgled stranice. Dok neke tvrtke i dalje žele imati zasebnu verziju svoje web stranice za mobilne korisnike, responzivni web dizajn postaje norma jer nudi veću svestranost pri nižim razvojnim troškovima.

Izrada web lokacije koja je u skladu s pravilima responzivnog web dizajna, traje znatno manje od izrade samostalne mobilne aplikacije uz standardnu internetsku stranicu radne površine. Čak i ako početno ulaganje responzivno dizajnirane web stranice završi, dolazi do toga da bude skuplje od stvaranja dvije zasebne web stranice, na kraju će se uštedjeti na dugoročnom planu zbog troškova održavanja. Neki od nedostataka responzivnog web dizajna su ta da treba više vremena kako bi se učitao sadržaj responzivne web stranice, jer su slike na responzivnom mjestu samo vizualno smanjene. Loša kompatibilnost sa starijom verzijom Internet Explorera. Potrebno je dosta vremena za razvoj responzivne web stranice. Ako se želi pretvoriti postojeću

web lokaciju na respozivnu web lokaciju, potrebno je malo više vremena. Ako se želi nova web stranica, bolje je krenuti s dizajnom od samoga početka.

2.2.3.Bootstrap

Kreiranje nove respozivne stranice često može biti obeshrabljujuće zbog mnogih problema koji se pojavljuju vezanih uz respozivan dizajn. Neki od tih problema su računanje kolumni, kreiranja tzv. mreže internetske stranice, računanja postotaka i kontrolnih točaka u dizajnu (točke koje koristimo pri određenoj širini preglednika koje kažu dizajnu da više ne prikazuje dvije kolumne, nego da druga npr. padne ispod prve) te još mnogo toga. Mnogi dizajneri stoga rješenje prolaze u već kreiranim i isprobanim „kosturima“ stranica. Kostur je skup unaprijed definiranih CSS pravila koja se mogu koristiti za brzi razvoj. Oni obično obuhvaćaju tipografiju, obrasce a često i predloške rasporeda. Trenutno je popularan kostur Bootstrap, koji je razvijen za potrebe Twittera te nudi brojne unaprijed oblikovane rasporede, tipografiju, opcije obrasca, određeni broj komponenata koje se mogu višekratno koristiti pa čak i proširivost putem JavaScripta. Proširenja obuhvaćaju varijable i ugniježđena pravila, što omogućava brži rad te prilagođene funkcije koje korisniku pružaju dodatnu snagu.

Bootstrap je programski okvir na korisničkoj strani, a razvijen je za potrebe Twittera od strane Marka Otta i Jacoba Thorntona, 2011. godine. Bootstrap je najpopularniji HTML, CSS i JavaScript okvir za razvoj respozivnih projekata na webu. Bootstrap čini brži i jednostavniji razvoj respozivnih web stranica. Izrađen je za ljude svih vještina, uređaje svih oblika i projekte svih veličina. Bootstrap jednostavno i učinkovito pomaže pri nizradi respozivnih web stranica i aplikacija s jednim kodom, od pametnih telefona do tableta, stolnih računala s CSS medijskim upitima.

2.3. Povijest web dizajna

Prva web stranica je objavljena prije približno 30 godina. Stvaranje web preglednika koji može prikazivati slike bio je prvi korak u web dizajnu. Najbliža opcija koja je bila dostupna za strukturiranje informacije, jest koncept tablica već postojećih u HTML-u. Iako se nije činila potpuno ispravna opcija, jer je glavna svrha tablice struktura brojeva, još uvijek je bila uobičajena metoda dizajna weba. Veliki problem je bio u održavanju tih struktura. Web dizajneri bi napravili odličan dizajn, a programeri bi ga rastavljali na sitnije dijelove i stavljali u kod. S druge strane, tablice su imale odlične značajke poput sposobnosti usklađivanja okomito, definiranja u pikselima ili postocima. Glavna prednost jest da je to bila najbliža verzija mreže koja danas postoji.

1996. godine na tržištu se pojavio Adobe Flash. Adobe Flash je multimedijaska softverska platforma koja se koristi za proizvodnju animacija, bogatih internetskih aplikacija, aplikacija za stolna računala, mobilnih aplikacija, mobilnih igara i ugrađenih video zapisa web preglednika. Flash prikazuje tekst, vektorsku grafiku kako bi osigurao animacije, video igre i aplikacije. Omogućuje strujanje zvuka i videozapisa.

Web se neprekidno mijenja, neke stvari će se promijeniti, neke ostati a neke odumrijeti potpuno. Verzija CSS-a koja se trenutno koristi je CSS2.1, što je revizija specifikacije CSS2 objavljene 1997. godine. U zadnjih nekoliko godina govori se o specifikaciji CSS3, čiji je rad započeo još 1998. godine. Godinu dana nakon što je objavljena specifikacija CSS2. Tijekom 2005. Svi CSS3 moduli vraćeni su u status nacrti i proces uređivanja je krenuo ispočetka. Ethan Marcotte 2010. godine predlaže da sadržaj na stranici ostaje isti, ali da se raspored elemenata mijenja. Tako je nastao responzivni web dizajn (eng. Responsive web design), čija je ideja da isti sadržaj funkcionira na različitim veličinama ekrana.

2.4.CMS

CMS ili Content Management System je sustav koji omogućuje upravljanjem sadržajem. CMS nam omogućava da u bilo kojem trenutku i s bilo kojeg mjesta možemo upravljati sadržajem internet stranica koje izrađujemo bez da posjedujemo znanje programiranja. Dovoljan je samo internet preglednik i poznavanje osnova korištenja računala. Sustav za upravljanjem sadržajem je softverska aplikacija ili skup povezanih programa koji se koriste izradu i upravljanje digitalnim sadržajem. Jasna je uloga CMS sustava kada je riječ o upravljanju sadržajem, ako se koristi CMS ne mora se ručno pisati HTML kod. Prednost korištenja CMS-a umjesto ručne izrade statičkih ili dinamičkih Web stranica je u tome što CMS omogućava jednostavnije i fleksibilnije unošenje promjena. Upravo to radi jedan od najpopularnijih softvera za vođenje blogova ili izradu stranice, WordPress. Postoji ogroman broj slobodnih i pretplatničkih CMS ponuda dostupnih za osobnu i poslovnu upotrebu. U nastavku su nabrojani neki primjeri poslužitelja platforme CMS:

SharePoint, Joomla, WordPress, Pulse CMS... Jedan od najpoznatijih CMS sustava je WordPress. WordPress je najjednostavniji i najpopularniji način stvaranja vlastite web stranice ili bloga. WordPress ima ovlasti u 29% web stranica na internetu. Na tehničkoj razini, WordPress je sustav za upravljanjem sadržajem otvorenog koda, što znači da svatko može mijenjati ili koristiti Wordpress softver besplatno. Sustav za upravljanje sadržajem je u osnovi alat koji olakšava upravljanje važnim aspektima web stranice, poput sadržaja, bez znanja programiranja. Krajnji rezultat je da Wordpress čini izgradnju web stranice dostupne svima, čak i ako niste programeri.

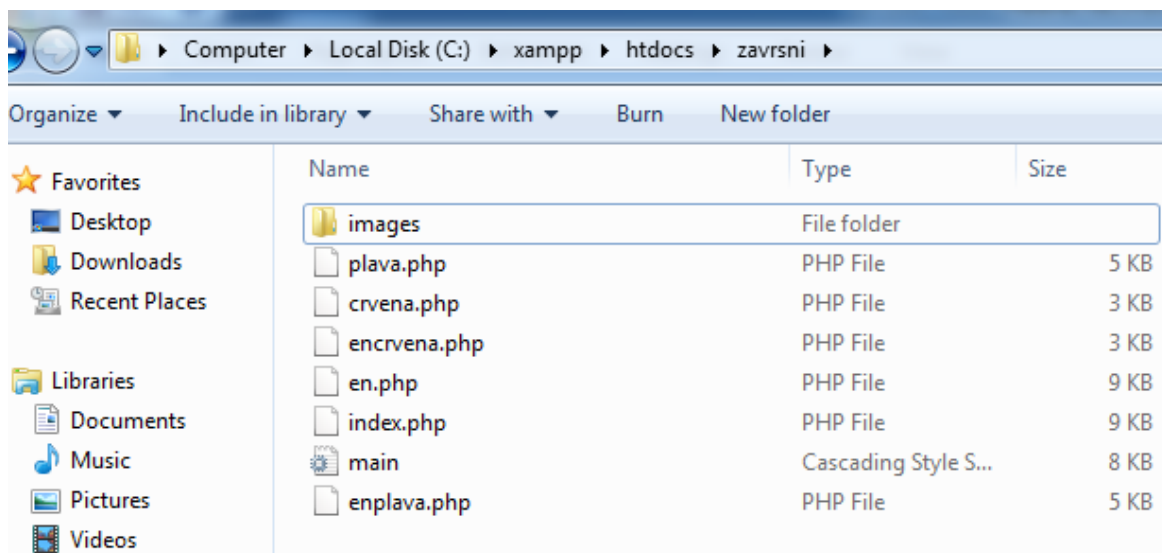
WordPress generira Web stranice korištenjem tri komponente: jezgre, teme i baze podataka. Jezgra je skup datoteka koje su preuzete s adrese wordpress.org i koje obavljaju posao spremanja, preuzimanja i sastavljanja sadržaja. Sadržaj se čuva u bazi podataka a tema je sastavljena od predložaka koji jezgri pružaju informacije o tome koji sadržaj treba preuzeti i sastaviti. Upravljanje sadržajem odnosi se na proces unosa sadržaja u bazu podataka, omogućavanje njegovog uređivanja i zatim čuvanje uputa o načinu na koji se taj sadržaj odnosi prema drugom sadržaju. WordPress pruža vrlo jednostavno i pregledno sučelje s kojim su zadovoljne stotine tisuća korisnika širom svijeta.

3. Praktični dio

Kao praktični dio rada, biti će prikazana web aplikacije „Apartmani Zaton“, responzivnog web dizajna. Odnosno rađena je uz pomoć Bootstrap programskog okvira, koji uz pomoć grid sistema značajno olakšava izradu responzivne web stranice. Web aplikacija je napravljena kako bi se prikazao osnovni sadržaj Apartmana „Zaton“, na što jednostavniji način. Stoga je napravljen jednostavan odabir između dvije raspoložive zgrade koja sadrži svoje apartmane i pritiskom na određenu zgradu (plavu ili crvenu) prikazane su fotografije apartmana sa svojim pojedinostima.

3.1. Opći prikaz web aplikacije

Za početak je potrebno kreirati mapu na računalu koja će sadržavati sve što će se nalaziti na internetskoj stranici i koja će biti postavljena na server. Podmape koje bi gotovo svaka stranica trebala sadržavati su css podmapa te podmapa sa slikama. Sve slike koje će se koristiti u dizajnu postavljaju se u podmapu images. Na poslijetku su kreirana šest PHP dokumenta za glavnu stranicu (index.php) i sve podstranice te CSS dokument koji se nalazi u css podmapu (main.css). Sadržaj mape je prikazan na slici 2.



Slika 2. Sadržaj mape.

Na samome početku projekta, potrebno je definirati stilski izgled same aplikacije, koja se kreira uz pomoć već spomenutih tehnologija HTML i CSS, pomoću čijeg će se doprinosa i prikazati cjelokupna responzivna web stranica. Cijelokupni izgled se bazira na HTML-u te povezivanjem CSS stilskih listova iz vanjske datoteke. Kao što je već spomenuto, stranica je responzivna stoga je potrebno umetnuti Bootstrap programski okvir u samu HTML datoteku. U nastavku će se prikazati sam postupak tijekom izrade web aplikacije, odnosno detalje iz koda koji će se naknadno i pojasniti kako bi se lakše objasnio postupak izrade.

Kodiranje stranice započinje index.php datotekom te definiranjem vrste dokumenta pomoću `<!DOCTYPE>` elementa. Zatim se otvara i zatvara `html` element unutar kojeg se isto tako otvaraju i zatvaraju `<head>` i `<body>` elementi. Unutar `<head>` elementa definira se naslov internetske stranice `<title>` elementom. Pomoću meta elemenata i raznih atributa i svojstava definiraju se vrsta znakova koja se koristi za kodiranje. Posljednji element koji se nalazi unutar `<head>` elementa jest `<link>` element kojim se HTML dokument upućuje na lokaciju CSS dokumenta u kojem je definirano oblikovanje stranice. Sav dosad definirani kod može se kopirati i primijeniti na sve .php dokumente (dakle na sve podstranice). Za razliku od `<head>` elementa, unutar `<body>` elementa nalazi se sav vidljivi sadržaj stranice. Stranica je raspodijeljena na header (vrh, glava stranice), navigaciju, sadržaj te footer (dno, podnožje stranice).

3.1.1. Dizajn zaglavlja

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="main.css">
    <link href='https://fonts.googleapis.com/css?family=Raleway:500,900,200' rel='stylesheet' type='text/css'>
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css"
    integrity="sha384-1q8mTJOASx8j1Au+a5WDVnPi2lkPfwEAa8hDDdjZlpLegxhjVME1fgjWPGmkzs7"
    crossorigin="anonymous">
    <title>Apartmani "Zaton"</title>
  </head>
```

Slika 3. Uključivanje Bootstrap programskog okvira u HTML.

Slika 3. prikazuje izgled naslovne web stranice u obliku koda, odnosno prikaz zaglavlja i umetanje osnovnih oznaka za početak izrade web stranice. Postavljena je oznaka za prikaz hrvatskih znakova, nakon toga element metapodataka `<meta>` koji daje upute pregledniku o tome kako upravljati dimenzijama stranice. `Width=device-width` uređaja određuje širinu stranice kako bi slijedila širinu zaslona uređaja (koja ovisi o samom uređaju). `Initial-scale=1.0` postavlja početnu razinu zumiranja nakon što preglednik prvi put učita stranicu. zatim poveznica na vanjsku CSS datoteku `main.css`, u kojoj se nalaze sve CSS upute koje su korištene u dokumentu. Nakon toga su uključeni Bootstrap poveznice za font i grid sustav pomoću kojeg će stranica dobiti responzivni izgled. Na kraju slike 3. nalazi se oznaka za naslov i definiran je naslov stranice.

3.1.2.Bootstrap

Bootstrap je besplatan programski okvir za programiranje na strani korisnika koji služi za brži i jednostavniji web razvoj. Postoje dva načina za početak korištenja programskog okvira Bootstrap na vlastitoj web stranici:

- Preuzeti Bootstrap sa stranice getbootstrap.com.
- Uključiti Bootstrap s CDN-a, mreža za isporuku sadržaja (engl.content delivery network).

MaxCDN pruža CDN podršku za Bootstrapov CSS i JavaScript. Na slici 4. je prikazano uključivanje maxcdn-a unutar head oznaka.

Bootstrap 3 je dizajniran da odgovara prikazu na mobilnim uređajima. *Mobile-first styles* dio su osnovnog okvira. Da bi se osiguralo pravilno prikazivanje i zumiranje na dodir, dodaje se `<meta>` oznaka unutar elementa `<head>` također prikazana na slici 4. Bootstrap također zahtijeva *container* element koji sadrži sadržaj web stranice.


```

<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="main.css">
    <link href='https://fonts.googleapis.com/css?family=Raleway:500,900,200' rel='stylesheet' type='text/css'>
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css"
    integrity="sha384-1q8mTJOASx8j1Au+a5WDVnPi2lkFfwwEAa8hDDdjZlpLegxhjVME1fgjWPGmkzs7"
    crossorigin="anonymous">
    <title>Apartmani "Zaton"</title>
  </head>

```

Slika 4. Bootstrap.

Bootstrap omogućuje do 12 stupaca preko stranice. Sljedeći primjer pokazuje kako dobiti tri stupca s jednakom širinom koja počinju na tabletima a završavaju na velikim stolnim računalima.

```

<div class="row">
  <div class="col-sm-4">.col-sm-4</div>
  <div class="col-sm-4">.col-sm-4</div>
  <div class="col-sm-4">.col-sm-4</div>
</div>

```

Prethodni primjer prikazan u HTML-u na stranici je na slici 5:

```

<div class="container">
  <div class="col-md-4">
    <a href="crvena.php"></a>
  </div>
  <div class="p6 col-md-4">Nudimo Vam izbor između dvije različite zgrade, sa različitim
  tipovima apartmana. U svakoj zgradi dostupan je pristup i kamenom roštilju koji se nalazi odmah ispred zgrade.
  Svaki apartman ima svoje parkirno mjesto.</br>
  <br> Apartmane možete vidjeti pritiskom na zgradu po odabiru. Unutar svake zgrade nalaze se apartmani
  na jednoj ili dvije etaže, u prizemlju ili na katu.
  </div>
  <div class="col-md-4">
    <a href="plava.php"></a>
  </div>
</div>

```

Slika 5. Bootstrap stupci.

Primjer različitih stupaca prikazan je na slici 6:

```
<div class="container-fluid no-padding">
  <div class="nav">
    <div class="col-md-3 col-xs-4">
      <p class="p">+38595 566 4198</p>
    </div>
  <div class="col-md-6 col-xs-4">
    <p class="p">Apartmani "Zaton"</p>
  </div>
  <div class="col-md-3 col-xs-4 nav2">
    <a href="https://www.facebook.com/">
      
    </a>
  <div class="col-md-1">
    <a href="index.php">
      HR
    </a>
  </div>
  <div class="col-md-1">
    <a href="en.php">
      EN
    </a>
  </div>
</div>
</div>
</div>
</div>
```

Slika 6. Različiti stupci.

3.1.3. Dizajn navigacije

Slika 4. prikazuje tijelo stranice sa svojim elementima i navigaciju. U kodu se nalaze div elementi koji su uređeni na eksplicitan način. U tijelu stranice prvo je stvoren spremnik (engl. Container) sa dodatnim svojstvom *fluid* koji pruža spremnik pune širine te obuhvaća čitavu širinu prozora. No-padding uklanja razmake između stupova u Bootstrapu. Nakon toga je otvoren div element u kojem će se nalaziti naslov stranice, kontakt broj i opcije za hrvatsku i englesku verziju. U slijedećem div elementu definirane su klase za prikaz na uređajima xs i md veličine odnosno za prikaz na mobilnim uređajima i stolnom računaru. Navigacija je podijeljena na tri stupca nejednake veličine (col-md-3, col-md-6, col-md-3). U prvom stupcu (col-md-3) nalazi se broj za kontakt u sredini naslov stranice te u trećem stupcu poveznica za facebook stranicu apartmana i opcije za jezike. Jezici su smješteni unutar zadnjeg stupca te su podijeljeni u vlastite stupce (col-md-12). Facebook poveznici je dodano svojstvo nav2, kako bi se centrirala u svome stupcu.

```

<body>
  <div class="container-fluid no-padding">
    <div class="nav">
      <div class="col-md-3 col-xs-6">
        <p class="p">+38595 566 4198</p>
      </div>
    <div class="col-md-6 col-xs-6">
      <p class="p">Apartmani "Zaton"</p>
    </div>
    <div class="col-md-3 nav2">
      <a href="https://www.facebook.com/">
        
      </a>
    <div class="col-md-1">
      <a href="index.php">
        HR
      </a>
    </div>
    <div class="col-md-1">
      <a href="en.php">
        EN
      </a>
    </div>
  </div>
</div>

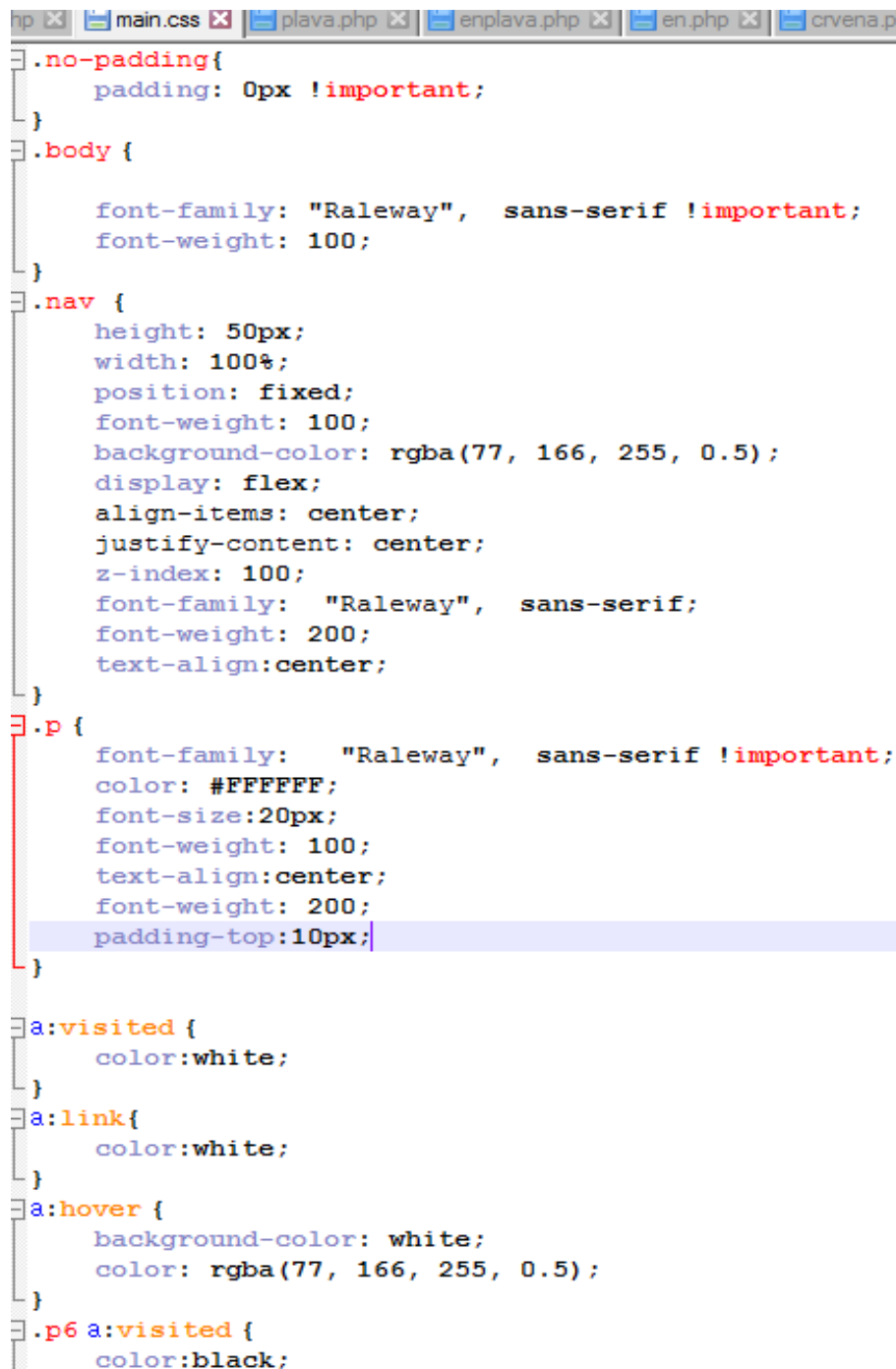
```

Slika 7. Prikaz body djela koda.

Slika 5. Prikazuje CSS kod, odnosno način na koji je uređena navigacija u CSS-u. Selektor no-padding uz sebe ima pravilo !important što je pravilo koje će se uvijek primjenjivati bez obzira gdje se to pravilo pojavljuje u CSS dokumentu. Također vrijedi i za body selektor, kojemu je dodijeljen font-family i težina fonta. Font-weight definira debljin fonta. Raspon je od 100 do 900. Normalni font je 400. 700 je podebljano. U primjeru je odabrana najmanja težina od 100. Visina navigacije je definirana u pikselina i ona iznosi 50px. Širina je sto posto, što znači da se proteže čitavom širinom ekrana. Pozicija je fiksna što znači da je element je postavljen u odnosu na prozor preglednika.

Boja je određena s RGBA. RGBA vrijednosti boje su proširenje RGB vrijednosti boje s alfa kanalom koji određuje neprozirnost boje. Vrijednost boje RGBA navedena je s: rgba (crvena, zelena, plava, alfa). Alfa parametar je broj između 0,0 (potpuno transparentan) i 1,0 (potpuno neproziran). Display:flex; širi elemente da bi popunili slobodni prostor ili ih smanjuje kako bi spriječio prelijevanje. Z-index, je CSS svojstvo koje određuje red stanja specifičnih elemenata. Element s većim redoslijedom uvijek je ispred drugog elementa s donjim redoslijedom. Što znači da će elementi u navigaciji biti iznad elemenata preko kojih se sadržaj u navigaciji kreće. Tekst u navigaciji je uređen pomoću selektora p. Boja je #FFFFFF odnosno bijela boja, RGB vrijednost je (255,255,255). Padding elementa je prostor između njegovog sadržaja i njegove granice. Padding-top postavlja gornji razmak elementa. Selektorom a, uređene su poveznice na drugi

dokument. `a:visited`, znači da će poveznice promijeniti boju nakon što su jednom posjećene, `a:link` selektor označava normalnu, neposjećenu poveznicu, `a:hover` predstavlja selektor koji u slučaju na slici mijenja boju u `rgba(77, 166, 255, 0.5)` nakon što se mišem pređe preko same poveznice i boju pozadine u bijelu.

The image shows a screenshot of a web browser's developer tools or a code editor. At the top, there are several tabs: 'hp', 'main.css', 'plava.php', 'enplava.php', 'en.php', and 'crvena.p'. The main content is a CSS code block with the following rules:

```
.no-padding{
  padding: 0px !important;
}
.body {
  font-family: "Raleway", sans-serif !important;
  font-weight: 100;
}
.nav {
  height: 50px;
  width: 100%;
  position: fixed;
  font-weight: 100;
  background-color: rgba(77, 166, 255, 0.5);
  display: flex;
  align-items: center;
  justify-content: center;
  z-index: 100;
  font-family: "Raleway", sans-serif;
  font-weight: 200;
  text-align:center;
}
.p {
  font-family: "Raleway", sans-serif !important;
  color: #FFFFFF;
  font-size:20px;
  font-weight: 100;
  text-align:center;
  font-weight: 200;
  padding-top:10px;
}
a:visited {
  color:white;
}
a:link{
  color:white;
}
a:hover {
  background-color: white;
  color: rgba(77, 166, 255, 0.5);
}
.p6 a:visited {
  color:black;
```

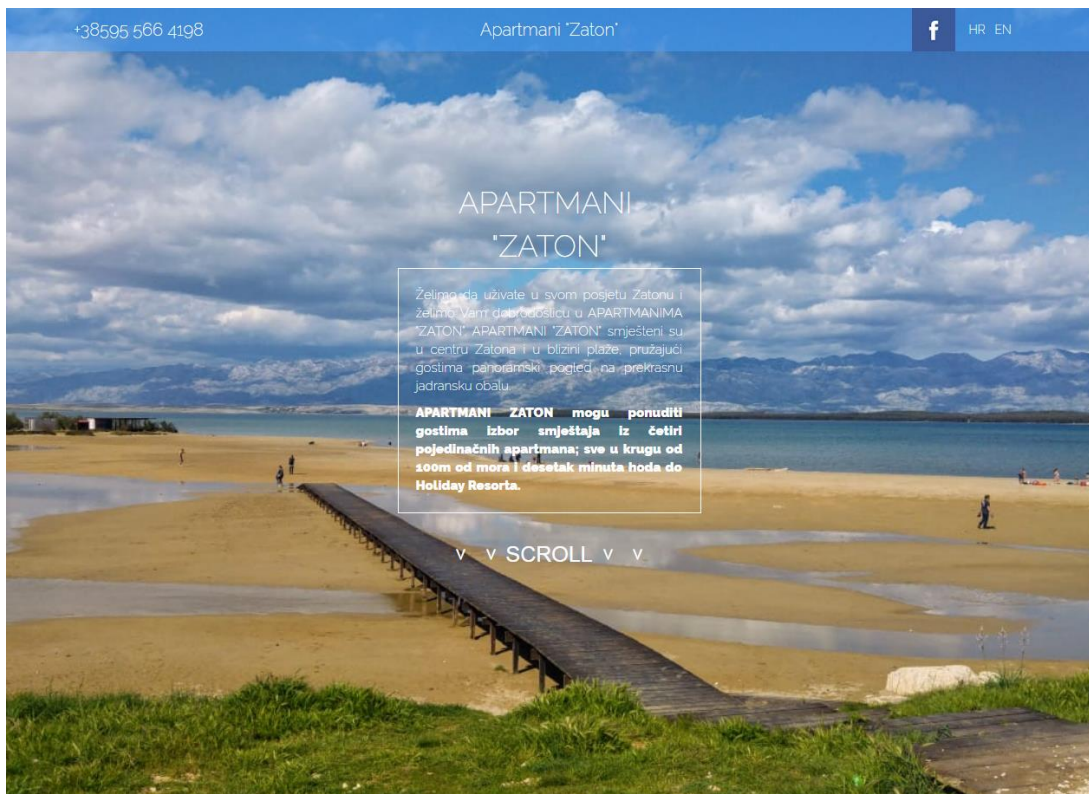
Slika 8. CSS navigacije.

3.1.4.Parallax

Parallax je slika preko koje se sav sadržaj ili samo određeni dio stranice stranice treba kretati. Slika za parallax dodana je u CSS-u pomoću svojstva background-image: url(images/IMG-20180818-WA0011.jpg); i prilagođena odgovarajućim svojstvima ekrana primjerice koja će biti visina slike u parallaxu (height:100vh;) te prikazuje jednu od plaža u Zatonu. Nakon toga postavljen je još jedan spremnik u kojem se nalazi „row“. Row je „omot“ za stupce grida. Svaki stupac ima vodoravnu podlogu za kontrolu razmaka između njih. Row-u je dodan atribut sa nazivom element koji sadržaj unutar spremnika pomiče niže za 200px; Odnosno gornja margina mu je definirana u CSS-u (margin-top:200px;). Parallax je podijeljen na tri jednaka stupca: col-md-4, col-md-4, col-md-4.

U srednjem stupcu je sadržaj koji se pomiče preko parallaxa, u sadržaju je naslov stranice koji se nalazi preko cijelog srednjeg stupca svojstvom col-md-12. Nakon toga, unutar linija se nalazi kratak tekst, od kojeg je prvi dio teksta svijetliji od drugog što je postignuto oznakom ** ** u samome HTML kodu.

Nakon zatvaranja tog spremnika, dodaje se tekst koji sadrži strelice koje su okrenute za 90 stupnjeva, svojstvom transform: rotate(90deg); u CSS-u. Nakon prikaza izgleda djela stranice u kodnom obliku te samog pisanja koda, dobiva se izgled prikazan na slici 6. :



Slika 9. Parallax.

3.1.5.Sadržaj stranice

Kao „razdvajač“ elemenata na stranici stvorena je linija, kako bi sadržaj bio ljepše uređen. Linija je postavljena izvan spremnika, u zasebnom <div> elementu. Linija je zamišljena tako da ide preko cijele širine prozora, što je postignuto atributom col-md-12. U CSS-u je uređena selektorom crta, kao što je prikazano na slici 7. Sama linija je stvorena tako da joj je određena granična linija od debljine 1 piksela, crne boje. Sa razmakom od gornjeg elementa 50 piksela.

```
.crta{  
  border:1px solid black;  
  position:fixed;  
  margin-top:50px;  
}
```

Slika 10. Linija.

Nakon linije, postavljene su dvije fotografije koje su ujedno i poveznica na drugi .php dokument. U prvom spremniku nakon spomenute linije nalazi se tekst koji je postavljen u samoj sredini stranice. Nakon teksta slijedi manja linija koja razdvaja dva teksta. Linija je napravljena tako da zauzima trećinu prozora, odnosno smještena u drugom stupcu (col-md-4). Razmak između gornjeg i donjeg sadržaja jest 10 piksela, što se postiglo selektorima margin-top i margin-bottom. Slika 8. Prikazuje prethodni opis.

```
.linija{  
  border:1px solid #33A8FF;  
  align-items:center;  
  justify-content:center;  
  text-align:center;  
  margin-top:10px;  
  margin-bottom:10px;  
}
```

Slika 11. Manja linija CSS.

Tekst je uređen u CSS-u selektorom p6. Definirana mu je boja, veličina fonta, poravnanje teksta, obitelj fonta sa pravilom !important, težina fonta i gornji razmak od 30 piksela što je prikazano na slici 11.

```
.p6 {  
  color:#2f2f1e;  
  font-size: 15px;  
  text-align: justify;  
  font-family: "Raleway", sans-serif !important;  
  font-weight: 100;  
  margin-top: 30px;  
}
```

Slika 14. Tekst CSS.

Sva tri stupca sa konačnim izgledom prikazana su na slici 12.

APARTMANI "ZATON" NALAZE SE U DVIJE
RAZLIČITE ZGRADE

NAJPOVOLJNIJA CIJENA,
U BLIZINI GRADA NINA I ZADRA



Nudimo Vam izbor između dvije različite zgrade, sa različitim tipovima apartmana. U svakoj zgradi dostupan je pristup i kamenom roštilju koji se nalazi odmah ispred zgrade. Svaki apartman ima svoje parkirno mjesto.

Apartmane možete vidjeti pritiskom na zgradu po odabiru. Unutar svake zgrade nalaze se apartmani na jednoj ili dvije etaže, u prizemlju ili na katu.



Slika 15. Stupci sa slikama i tekстом konačan izgled.

3.1.6.Slideshow

Slideshow je prezentacija nekoliko fotografija jedne za drugom pritiskom na dugme za sljedeću fotografiju. Slider prikazuje neke od plaža u Zatonu. Slideshow je smješten unutar spremnika pune širine koji obuhvaća čitavu širinu prozora. Nalazi se u sredini spremnika, što je definirano u CSS-u. Slider je napravljen uz pomoć HTML, CSS i JavaScripta.

3.1.6.1.HTML

U HTML-u su definirane fotografije koje će se prikazati u slideshow-u. Fotografija koja je odabrana je prikazana, a ostale se nalaze ispod nje. Slideshow je pozicioniran unutar jednog stupca čija je veličina col-md-6. Definirani su redni brojevi pojedine slike koja je otvorena u trenutku. Te linkovi u obliku strelice, koji pritiskom na njih odlaze na sljedeću ili prethodnu fotografiju u slideshow-u. Unutar tog istog spremnika nalazi se div element sa atributom: "red". U njemu se nalazi prikaz ostalih fotografija u obliku manjih sličica, koje nisu odabrane. Prikaz prethodnog opisa HTML koda, bez prikaza manjih sličica, nalazi se na slici 12. :

```
<div class="container-fluid no-padding">
  <div class="sadržaj">
    <div class="slider col-md-6">
      <div class="mySlides">
        <div class="numbertext">1 / 6</div>
        
      </div>
      <div class="mySlides">
        <div class="numbertext">2 / 6</div>
        
      </div>
      <div class="mySlides">
        <div class="numbertext">3 / 6</div>
        
      </div>
      <div class="mySlides">
        <div class="numbertext">4 / 6</div>
        
      </div>
      <div class="mySlides">
        <div class="numbertext">5 / 6</div>
        
      </div>
      <div class="mySlides">
        <div class="numbertext">6 / 6</div>
        
      </div>
      <a class="prev" onclick="plusSlides(-1)">#10094;</a>
      <a class="next" onclick="plusSlides(1)">#10095;</a>
    </div>
  </div>
</div>
```

Slika 16. Slideshow HTML.

3.1.6.2.JavaScript

JavaScript skripta postavljena je unutar body oznake. Dio JavaScript koda, pomoću kojega je napravljen slideshow na naslovnoj stranici je prikazan u nastavku:

```
<script>

var slideIndex = 1;

    showSlides(slideIndex);

        function plusSlides(n) {
            showSlides(slideIndex += n);
        }

        function currentSlide(n) {
            showSlides(slideIndex = n);
        }

        function showSlides(n) {
            var i;
            var slides =
document.getElementsByClassName("mySlides");

            var dots =
document.getElementsByClassName("demo");

            var captionText =
document.getElementById("caption");

            if (n > slides.length) {slideIndex = 1}
            if (n < 1) {slideIndex = slides.length}

            for (i = 0; i < slides.length; i++) {
                slides[i].style.display = "none";
            }

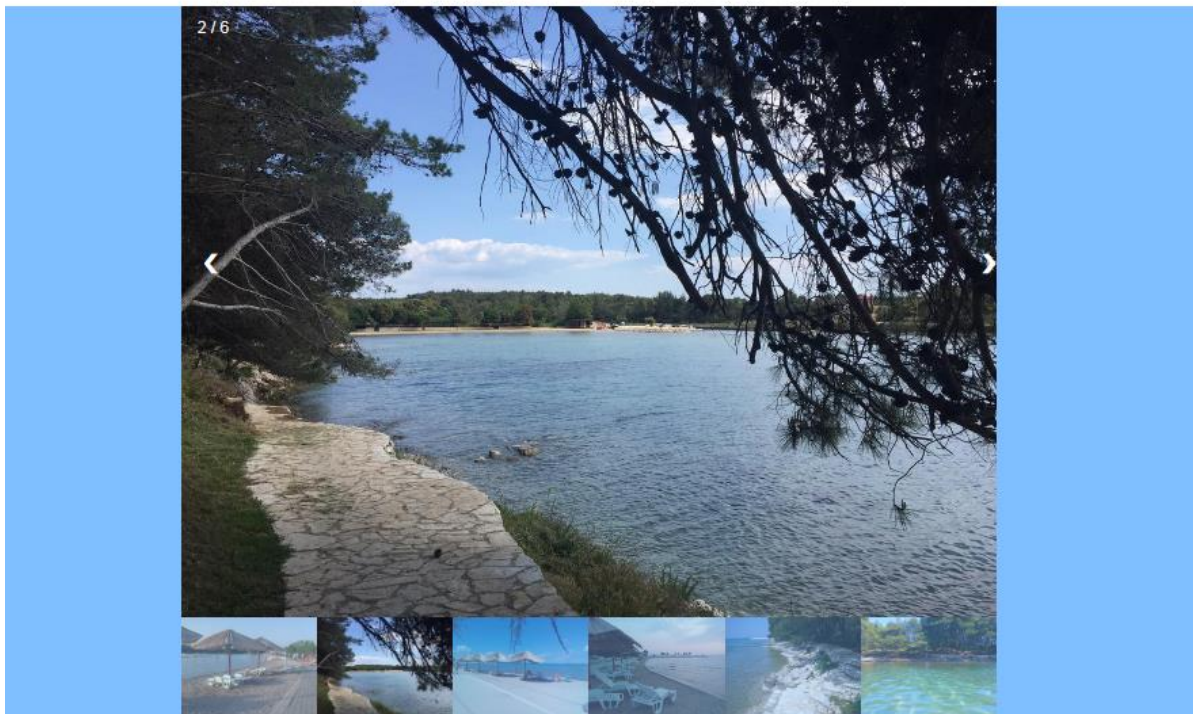
            for (i = 0; i < dots.length; i++) {
                dots[i].className = dots[i].className.replace(" active", "");
            }

            slides[slideIndex-1].style.display = "block";
```

```
dots[slideIndex-1].className += " active";  
captionText.innerHTML = dots[slideIndex-1].alt;  
    }    </script>;
```

3.1.6.3.Konačan izgled slideshow-a

Na slici 13. prikazan je krajnji izgled spoja tri jezika, odnosno slideshow onako kako je postavljen na samoj stranici. Slideshow se nalazi na plavoj boji pozadine, što je uređeno u CSS-u, koja također prevladava na samoj stranici. U slideshow-u se nalaze fotografije plaža i njihove dimenzije su prilagođene samom slideshow-u.



Slika 17. Konačan izgled slideshow-a.

3.1.7.Google Map

U novo otvorenom spremniku nalazi se okvir u kojemu je smještena Google karta sa točnom lokacijom „Apartmana Zaton“. U drugom stupcu nalaze se osnovni podaci s kontaktima. Naslov „Kako do nas“ je smješten u sredini spremnika i zauzima svih 12 stupaca. U CSS-u je uređen tekst, odnosno u vanjskoj datoteci main.css. Nakon toga otvoren je div element s prvom polovicom spremnika u kojemu se nalazi *iframe* s Google kartom. Zatim su postavljena dva

prazna stupca kako bi napravila dovoljan razmak, a karta je smještena u osam stupaca. Google Karte olakšavaju ugradnju karte na web stranicu. Međutim, prema zadanim postavkama Google karte ne pružaju odgovarajuću responzivnost. Stoga je dodan atribut *map-responsive* kako bi to učinio mogućim. Prvo je potrebno u pregledniku otići na Google Maps, zatim je potrebno upisati ili pronaći željeno područje ili adresu koja se želi prikazati u okviru. Zatim se generira kod sa odgovarajućom upisanom adresom, kopira se taj isti kod i postavi unutar iframe elementa. Koji je smješten unutar div elementa sa klasom *map-responsive*. HTML kod je prikazan na slici 15.:

```
<div class="container-fluid no-padding">
  <div class="boja">
    <div class="row">
      <div class="col-md-12 pl b">KAKO DO NAS</div>
      <div class="col-md-6">
        <div class="col-md-2"></div>
        <iframe src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d2859.1951196714635!2d15.163966915845664!3d44.22364107910591!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.1!3m3!1m2!1s0x4761f5ed8b8bbdb83A0x754b9af19fe88889!2sApartmani+Vrulja%2C+Zaton%2C+Zadar!5e0!3m2!1shr!2shr!4v1535112654608" width="400" height="300" frameborder="0" style="border:0" allowfullscreen></iframe>
      </div>
    </div>
  </div>
```

Slika 18.HTML Iframe

CSS koji čini tu kartu responzivnom izgleda kao na slici 16.

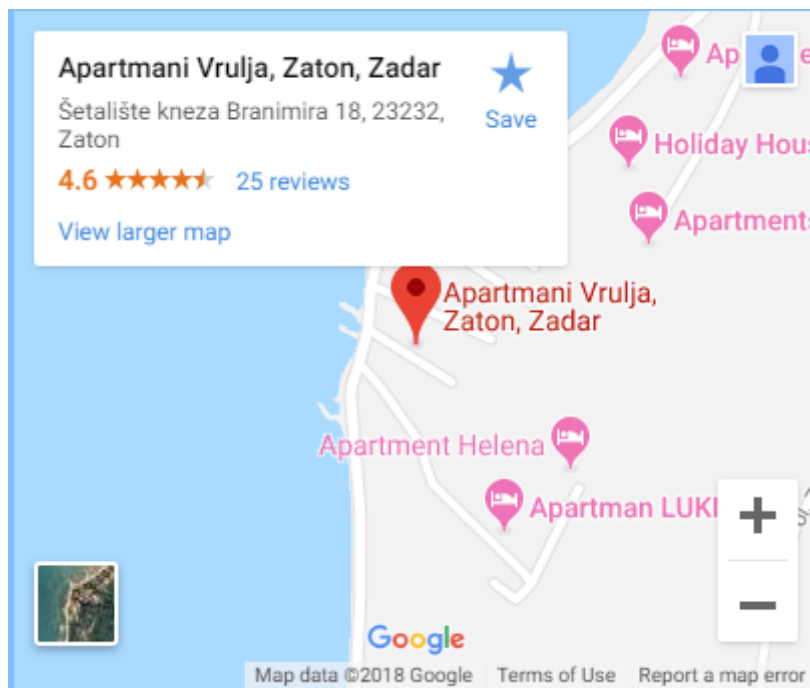
```
.map-responsive {
  overflow: hidden;
  padding-bottom: 56.25%;
  position: relative;
  height: 0;
}

.map-responsive iframe {
  left: 0;
  top: 0;
  height: 100%;
  width: 100%;
  position: absolute;
}
```

Slika 19. CSS Iframe

Selektoru map-responsive dodano je svojstvo *overflow* sa vrijednošću *hidden*. Svojstvo *overflow* određuje što se događa ako sadržaj prekorači okvir elementa. Što bi u ovom slučaju značilo da je sadržaj skriven. *Overflow* svojstvo se koristi kada se želi bolje kontrolirati izgled. CSS svojstvo *padding* se koristi kako bi se definirao prostor oko sadržaja elementa, unutar bilo koje definirane granice. U primjeru je svojstvo *padding-bottom* sa vrijednošću 56.25%. Specificira razmak u postocima ukupne širine elementa koji sadrži. Svojstvo *position* određuje vrstu metode pozicioniranja koja se koristi za element (statična, relativna, apsolutna, fiksna ili ljepljiva). U slučaju Google karte ta metoda je relativna i to znači da je postavljena relativno prema najbližem predaku (umjesto da je postavljena u odnosu na okvir za prikaz, kao što je fiksna pozicija). Međutim, ako apsolutni pozicionirani element nema postavljenih predaka, koristi tijelo dokumenta i kreće se pomicanjem stranice. Selektor map-responsive za *iframe* sadrži svojstvo *left* ili lijevo sa vrijednosti 0. Što znači da utječe na vodoravni položaj postavljenog elementa. Ovo svojstva nema utjecaja na elemente kojima nisu postavljene pozicije. Ako je položaj apsolutni kao u primjeru na slici 16. ili je fiksni, svojstvo *left* postavlja lijevi rub elementa u područje lijevo od lijevog ruba najbližeg predaka. Ako je položaj relativan, *left* svojstvo postavlja lijevi rub elementa na područje lijevo ili desno od normalnog položaja. Ako je pozicija definirana kao *sticky*, svojstvo *left* ponaša kao da je njegov položaj relativan kad je element unutar vidnog polja i kao da je njegov položaj fiksiran kad je izvan. U slučaju da je pozicija statična svojstvo *left* nema nikakvog učinka. Sljedeće svojstvo unutar selektora map-responsive jest *top* sa vrijednosti 0. Svojstvo *top* ili vrh utječe na vertikalni položaj postavljenog elementa. Ova svojstva nema utjecaja na elemente kojima nisu definirane pozicije kao i svojstvo *left*. Ako je pozicija apsolutna kao u navedenom primjeru ili je fiksna svojstvo *top* postavlja gornji rub elementa na područje iznad ili ispod gornjeg ruba najbližeg predaka. Ako je položaj relativan, *top* svojstvo postavlja gornji rub elemenata da se premjesti iznad ili ispod normalnog položaja. U slučaju da je pozicija definirana *sticky*, svojstvo *top* ponaša kao da je njegov položaj relativan u odnosu na element unutar prozora i kao da je položaj fiksiran kada je izvan. Ako je položaj statičan svojstvo *top* nema nikakvog učinka. Visina i širina su postavljene na 100%. A pozicija kao što je već spomenuto, je apsolutna.

Konačan izgled Google Map okvira prikazan je na slici 17.:



Slika 20. Google map.

U drugoj polovici div elementa nalaze se informacije sa osnovnim podacima za kontaktiranje i nekim korisnim poveznicama. Na slici 18. prikazan je HTML kod drugog stupca. U svakom redu nalazi se jedan podatak koji zauzima cijelu širinu svog stupca col-md-12.

```
<div class="col-md-6">
  <div class="row d">
    <div class="col-md-12 input">KONTAKTI </div>
    <div class="col-md-12 input">Adresa: </div> Apartmani Zaton </br> Ulica od Vrulje </br> Zaton </br> Hrvatska </br>
    <div class="col-md-12 input">Telefon:</div> GSM: +385 95 566 4198 </br> TEL: +385 1 2912 165 </br> TEL: +385 1 2930 965 </br>
    <div class="col-md-12 input">E-mail:</div> lukyhec@gmail.com <br>
    <div class="col-md-12 input">Korisni linkovi:</div> <a href="https://www.htz.hr/hr-HR">Hrvatska turistička zajednica</a></br>
    <a href="https://www.jadrolinija.hr/">Jadrolinija</a></br>
    <a href="http://meteo.hr/index.php">Prognoza</a>
  </div>
</div>
```

Slika 21. Kontakti.

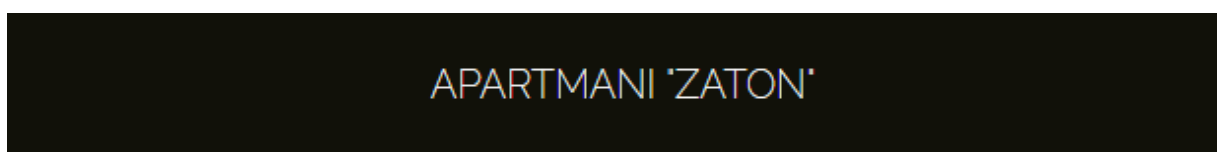
Konačan izgled stupca sa osnovnim informacijama je prikazan na slici 19.



Slika 22. Kontakti.

3.1.8.Podnoņje

Dno, podnoņje stranice (engl. Footer), nalazi se jedino na index.php stranici i na engleskoj inačici stranice. Na obje stranice footer je jednak. Footer je jednostavan, sadrži samo ime stranice koja je ujedno i poveznica na povratak na početak stranice, odnosno postavljena je da pritiskom na poveznicu, dovede na parallax. Podnoņje stranice prikazano je na slici broj 20.



Slika 23.Footer.

3.1.9.Podstranice

Sadržaj podstranica je jedini dio koji se razlikuje pošto su zaglavlje i navigacija na ostalim podstranicama jednaki. Ukoliko se dobro definiraju željene klase, iste se mogu koristiti na svim podstranicama što uvelike umanjuje vrijeme potrošeno na deklariranje CSS-a, te je jedini preostali posao ubacivanje sadržaja unutar div elementa. Sadržaj stranice nalazi se unutar div elementa koji ima klasu container i klase za određivanje stupaca npr.col-md.12. (vidljivo na slici 21.). Osim toga potrebno je deklarirati i paragrafe (p element), podebljan tekst (b) te div elemente koji će sadržavati prezentacijske radove u obliku slika. Zbog specifične pozicije tih div elemenata kreirano je nekoliko klasa kojima se oni korektno pozicioniraju na stranici. Na stranici se nalaze poveznice pojedinih apartmana koji vode na mjesto unutar istog dokumenta. Slika 23. prikazuje prethodno opisan kod. Na slici 21. je prikazan izgled internetske stranice u pregledniku.

+38595 566 4198

Plava zgrada

f HR EN HOME

U PLAVOJ ZGRADI NALAZE SE TRI APARTMANA

NAJPOVOLJNIJA CIJENA,
U BLIZINI GRADA NINA I ZADRA



Nudimo Vam izbor između tri različita apartmana. Sva tri apartmana su na dvije etaže, dvije terase (glavna/veća i manja/iz spavaće sobe). Apartmani imaju svoje parkirno mjesto i pristup roštilju.

Samo je jedan u prizemlju ostala dva su na prvom katu. Svaki apartman ima svoj posebni ulaz, bez ometanja ostalih stanara.

APARTMAN 1.

APARTMAN 2.

APARTMAN 3.

Slika 24. Izgled stranice u web pregledniku.

Slika 22. Prikazuje izgled stranice u pregledniku sa mobilnog uređaja (iPhone X):



U PLAVOJ ZGRADI NALAZE SE SE TRI APARTMANA

NAJPOVOLJNIJA CIJENA,
U BLIZINI GRADA NINA I ZADRA



Nudimo Vam izbor između tri različita apartmana. Sva tri apartmana su na dvije etaže, dvije terase (glavna/veća i manja/iz spavaće sobe). Apartmani imaju svoje parkirno mjesto i pristup roštilju.

Samo je jedan u prizemlju ostala dva su na prvom katu. Svaki apartman ima svoj posebni ulaz, bez ometanja ostalih stanara.

APARTMAN 1.

APARTMAN 2.

Slika 25. Izgled stranice sa mobilnog uređaja.

```

<div class="container">
  <div class="p5 col-md-12"> NAJPOVOLJNIJA CIJENA,</br>U BLIZINI GRADA NINA I ZADRA</div>
</div>
<div class="container">
<div class="col-md-4">
  <a href="plava.php"></a></div>
<div class="p6 col-md-4">Nudimo Vam izbor između tri različita apartmana.
Sva tri apartmana su na dvije etaže, dvije terase (glavna/veća i manja/iz spavaće sobe).
Apartmani imaju svoje parkirno mjesto i pristup roštilju. </br>
</br> Samo je jedan u prizemlju ostala dva su na prvom katu.
Svaki apartman ima svoj posebni ulaz, bez ometanja ostalih stanara. </div>
<div class="col-md-4">
  <p class="p6">
    <a href="#prvi">APARTMAN 1.</a></p>
  <p class="p6">
    <a href="#drugi">APARTMAN 2.</a></p>
  <p class="p6">
    <a href="#treći">APARTMAN 3.</a></p>
</div>
<div class="crta col-md-12"></div>
<div class="container-fluid no-padding">
  <div id="prvi">
    <div class="col-md-12 razmak">
      <div class="col-md-4"></div>
      <div class="col-md-4"></div>
      <div class="col-md-4"></div>
    </div>
  </div>
  <div id="prvi">
    <div class="col-md-12 razmak">
      <div class="col-md-4"></div>
      <div class="col-md-4"></div>
      <div class="col-md-4"></div>
    </div>
  </div>
</div>
<div class="crta col-md-12"></div>
<div class="container">
  <div id="drugi">
    <div class="col-md-12 razmak">
      <div class="col-md-4"></div>
      <div class="col-md-4"></div>
      <div class="col-md-4"></div>
    </div>
  </div>
  <div id="drugi">
    <div class="col-md-12 razmak">
      <div class="col-md-4"></div>
      <div class="col-md-4"></div>
      <div class="col-md-4"></div>
    </div>
  </div>
</div>
</div>

```

Slika 26. Podstranica.

3.2. Postavljanje na server i testiranje

Web stranica mora biti prilagodljiva jer web preglednici i računala koje koriste naši posjetitelji razlikuju se od onih koje mi koristimo te bi se učitavanje stranice u web pregledniku trebalo odvijati u razumnom vremenu jer ni propusnost mreža nije ista. Dakle, kriteriji koji se moraju provjeriti u ovoj fazi su: Sadržaj, izgled stranice, navigacija, originalnost, prilagodljivost ostalim web preglednicima, vrijeme učitavanja u web pregledniku. Navedeni kriteriji testiranja spadaju i u kriterije kojima se utvrđuje kvaliteta web stranica. Testiranje se može provesti automatski i ručno na temelju tehničke dokumentacije (uputama za testiranje) te ukoliko se pojave pogreške potrebno ih je odmah ispraviti. Kada je završena izrada web stranice, potrebno ju je objaviti na webu, učiniti dostupnom svima. U ovoj fazi potrebno je sve HTML i priložene datoteke (slike i sl.) pohraniti na slobodan prostor koji nam je omogućio naš web domaćin ili davatelj slobodnog prostora na poslužitelju. Postoje dva načina pohranjivanja datoteka web stranice na poslužitelju.

Jedan od načina jest uporaba grafičkog sučelja (engl. Web-based interface) koje neki web domaćini nude svojim korisnicima. Ako web domaćin ima takve mogućnosti, sve upute za korištenje takvog sučelja možemo pronaći na njegovoj službenoj web stranici. Web grafička sučelja su oblikovana na način da su razumljiva svim korisnicima weba bez obzira na njihovo iskustvo s webom. Drugi način, ujedno i način koji je korišten u ovom projektu jest uporaba programskog alata za prijenos datoteka s korisničkog računala na računalo poslužitelja preko interneta tzv. FTP programi. Svaki FTP program se spaja s poslužiteljem koristeći FTP protokol (engl. File Transfer Protocol) protokol za prijenos datoteka. Na slici 24. prikazana je index.php stranica otvorena u web pregledniku s mobilnog uređaja.



Slika 27. Iphone6.

4. Zaključak

Responzivni web dizajn ili RWD tehnika je koja olakšava prilagodbu web stranica različitim veličinama ekrana, odnosno rezolucijama prikaza. Nastala je iz potrebe da se poboljša korisničko iskustvo pregleda web stranice na mobilnom uređaju. Cilj ovog rada bio je napraviti responzivnu web stranicu, pomoću aktualnih inačica HTML i CSS tehnologija. Praktičnim dijelom dokumentacije prikazani su i dijelovi pojedinih opisnih sadržaja prvenstveno početak koji čini web programiranje opisom HTML i CSS stilskog jezika. Ova dva jezika čine temelj web programiranja i web rješenja. No, pojavom JavaScripta i sve češćom realizacijom istog stvaraju se novi putevi unutar korištenja različitih tehnologija. Time je uz pomoć JavaScripta napravljena tehnologija po imenu *Bootstrap* koja vrlo lako osvaja svijet s prednostima koje nudi. Od samoga responzivnog dizajna do dinamičnog sadržaja. S *Bootstrapom* sam se prvi puta susrela na stručnoj praksi, iako mi je na prvu izgledao kao vrlo komplicirano rješenje, dajljim učenjem i radom na projektima sam shvatila kako je to zapravo vrlo jednostavan programski okvir koji svatko uz malo truda i praktičnog rada može savladati. Samu temu završnog rada sam i izabrala iz razloga što sam već bila upoznata s time i izradila već nekoliko responzivnih stranica. Navela bih nekoliko stvari koje mi se ne sviđaju kod *Bootstrapa*, a to je potreba za ponovnim pisanjem stilova koji mogu oduzeti dodatno vrijeme provedeno na dizajniranju i kodiranju web stranice, u slučaju da dizajn odstupa od uobičajenog dizajna koji se koristi u *Bootstrapu*. Ono što volim kod *Bootstrapa* jest što je lagan i lako mu se prilagoditi, u slučaju rada na nekim većim projektima postoji mnogo besplatnih i profesionalnih predložaka. Iako postoji mnogo besplatnih alata i stranica koje nude već unaprijed gotove predloške, ovaj rad je u potpunosti pisan od početka. Kreirana stranica je testirana na drugim internetskim stranicama koje su specijalizirane za simulaciju responzivnosti na modernim uređajima manjih rezolucija npr. <http://www.responsinator.com>. te u različitim preglednicima na mobilnim uređajima (Samsung Galaxy S7, Samsung Galaxy S5, iPhone 5) i osobnim računalima. Sav sadržaj se uvijek prilagodio sukladno sa širinom zaslona.

Literatura

- [1] Crespo G, *Responzivni Web dizajn uz jQuery* (2010). IT expert.
- [2] Gasston P, *CSS3. FOOTPRINT*.
- [3] Gasston P, *Responzivni web dizajn uz HTML, CSS3 i Javascript*. FOOTPRINT.
- [4] Sochacki T, *JavaScript*. Helion.
- [5] Beaird J, *Načela dobrog web dizajna : smjernice za dobar web dizajn*. Dobar plan.
- [6] Worsley T, *Izrada web stranica*. Marjan tisak.
- [7] Tatroe K, MacIntyre P, Lerdorf R, *Programiranje PHP*. IT expert.
- [8] Plumpley G, *WordPress*. IT expert.
- [9] Galić D, *Izrada web stranica*. Element.
- [10] <https://www.w3schools.com/bootstrap/>
- [11] <https://www.w3schools.com/css/>
- [12] <https://www.w3schools.com/html/>

Popis slika

| | |
|------------------------------------------------------------------------|----|
| <i>Slika 1. Rezultat primjene HTML koda za tablicu</i> | 5 |
| <i>Slika 2.Sadržaj mape</i> | 16 |
| <i>Slika 3. Uključivanje Bootstrap programskog okvira u HTML</i> | 17 |
| <i>Slika 4. Bootstrap</i> | 19 |
| <i>Slika 5. Bootstrap stupci</i> | 19 |
| <i>Slika 6. Različiti stupci</i> | 20 |
| <i>Slika 7. Prikaz body djela koda</i> | 21 |
| <i>Slika 8. CSS navigacije</i> | 22 |
| <i>Slika 9. Parallax</i> | 23 |
| <i>Slika 10. Linija</i> | 24 |
| <i>Slika 11. Manja linija CSS</i> | 24 |
| <i>Slika 12. Manja linija</i> | 25 |
| <i>Slika 13. Stupci sa slikama i tekстом HTML</i> | 25 |
| <i>Slika 14. Tekst CSS</i> | 26 |
| <i>Slika 15. Stupci sa slikama i tekстом konačan izgled</i> | 26 |
| <i>Slika 16. Slideshow HTML</i> | 27 |
| <i>Slika 17. Konačan izgled slideshow</i> | 29 |
| <i>Slika 18.HTML Iframe</i> | 30 |
| <i>Slika 19. CSS Iframe</i> | 30 |
| <i>Slika 20. Google map</i> | 32 |
| <i>Slika 21. Kontakti</i> | 32 |
| <i>Slika 22. Kontakti</i> | 33 |
| <i>Slika 23.Footer</i> | 33 |
| <i>Slika 24. Izgled stranice u web pregledniku</i> | 34 |
| <i>Slika 25. Izgled stranice sa mobilnog uređaja</i> | 35 |
| <i>Slika 26. Podstranica</i> | 36 |
| <i>Slika 27. Iphone6</i> | 38 |